https://github.com/iamudyavar/amazon-review-analysis

Overview

In today's digital age, online shopping platforms, such as Amazon, have become essential for consumers to make informed decisions about products. However, a common challenge users face is having to sift through an overwhelming number of reviews to get a clear and concise understanding of what a product actually offers, how it performs, and whether it aligns with their preferences. This can be time-consuming and inefficient, often leaving users frustrated with the sheer volume of information available.

Our solution is aimed at improving the review exploration process for potential buyers. We set out to summarize product reviews, extract sentiment-based pros and cons, and analyze gender distribution based on review author names. This approach not only aids customers in making faster decisions but also provides vendors with valuable demographic insights about their products.

Transformer-based Review Summarization

Our core approach to summarizing the reviews involved building a transformer model from scratch, inspired by the "Attention is All You Need" paper, which introduced the concept of the transformer architecture [1]. This model allowed us to address the problem of summarizing long and complex reviews by capturing the relationships between words in a highly parallelizable manner.

To build our transformer model, we implemented the architecture described in the paper using PyTorch, which provided a flexible framework for defining custom modules such as multi-head self-attention and feedforward layers. The model includes self-attention mechanisms to weigh different parts of the input sequence based on relevance, positional encoding to retain the order of words, and an encoder-decoder structure to transform input (reviews) into output (summaries).

The results were strong: our model was able to compress lengthy, detailed customer reviews into concise summaries, often reducing the input text length by up to 90%. This level of compression retained the core sentiment and key information, making the summaries useful for product insights.

While our final model achieved great results, we encountered several challenges along the way. The main challenge was generating coherent summaries. Since our transformer was trained from scratch using only Amazon review data, the model occasionally produced incoherent outputs due to noisy and inconsistent input text. To mitigate this, we implemented a robust preprocessing pipeline using SpaCy and NLTK that included lowercasing, contraction expansion, stopword removal, elimination of non-alphabetic characters, and filtering out short or low-information words. These steps allowed the model to focus on the most informative parts of each review and improved both the training efficiency and output quality.

Additionally, we optimized the model architecture specifically for summarization (rather than translation, which was the original use case), leading to more coherent and concise outputs. Another challenge was long training times, which we addressed by reducing model complexity, shortening input and output sequence lengths, and using PyTorch's DataLoader for efficient batching. These changes significantly decreased training time without compromising performance.

Despite these challenges, the final transformer-based model significantly improved the ability to summarize large amounts of review data into concise, useful outputs

Sentiment and Part-of-Speech Analysis for Pros and Cons

The next component of our solution involved extracting sentiment-based pros and cons from the reviews. We initially began with sentiment analysis, where the goal was to classify each review as positive or negative. However, we encountered several challenges and pivoted to using the “Rating” column in the dataset as a stand-in for sentiment. For the final version, we decided to use part-of-speech tagging on the “Summary” data column of the reviews.

The first challenge we encountered was choosing the right technique. After experimenting with several models such as BERT and VADER, we found that sentiment analysis struggled with nuanced expressions, and often, it misclassified reviews with mixed sentiments. This led to many cases where low-rated reviews were classified as positive, and cons were misclassified as pros. To address this, we decided to use the rating itself as an alternative to sentiment. If a review had a high rating, we assumed that the sentiment towards the mentioned aspects was generally positive, and if the rating was low, the sentiment was likely negative. Though we briefly considered delving into aspect-based sentiment analysis, initial testing was unsuccessful and we decided to stand by the ratings and shift our focus to other components of the project.

The next challenge was deciding which parts of speech to focus on. Nouns often included irrelevant entities, such as the name of the product or "my dogs" in a review about dog food. On the other hand, some adjectives (e.g., "good," "amazing," "bad") were too vague and not descriptive enough to inform users about the product. We fine-tuned our extraction process to filter out common adjectives and focus on those that were more descriptive.

By addressing these challenges, we were able to create a refined list of the top 5 pros and cons from the review summaries, giving potential buyers a clearer understanding of the strengths and weaknesses of a product.

Gender Distribution Analysis

In addition to sentiment and summarization, we implemented a model to analyze the gender distribution of reviewers based on their usernames. This feature provides valuable insights into which demographic group tends to favor a product, helping buyers and vendors alike understand the product's intended audience.

We used a gender prediction model to classify the gender of reviewers based on their names [2]. While this approach was generally effective, we encountered issues with false positives, particularly with names that could be gender-neutral or ambiguous, due to some users having usernames that didn't resemble real names. There were also some issues with inaccuracies for certain regions, cultures, or gender identities. We addressed this by experimenting with different thresholds for the gender distribution for the model's outputs: the product is more popular among males, females, or equally popular among both genders. By analyzing the data and estimating an error bound for the false positives, we settled on having the model say a product would be equally popular among both genders at a 40/60 split, and skewed to one gender if it was any more divided than that.

Ultimately, despite some inaccuracies, the gender distribution analysis allowed us to provide a more comprehensive picture of which gender groups were more likely to leave reviews and share their opinions on a given product.

Conclusion

Our NLP project successfully addressed the challenge of navigating through a large volume of Amazon reviews by leveraging transformer-based summarization, sentiment analysis, and demographic insights. By building a transformer model from scratch, experimenting with various sentiment extraction techniques, and incorporating gender distribution analysis, we were able to create a system that significantly reduces the time spent by users browsing through reviews.

This solution benefits potential buyers by providing them with summaries and sentiment-driven pros and cons, and it also offers both them and product vendors valuable insights into the customer base of a product and the surrounding demographic patterns. Despite encountering challenges in training time, model accuracy, and data ambiguity, our project demonstrates the potential of NLP to enhance the online shopping experience and provide actionable insights for both consumers and vendors alike.

Contributions

Aditya: Worked on the base transformer model and built it up from scratch.
80 points - Significant exploration beyond baseline: Implemented the initial transformer architecture, and combined all team member's contributions into one pipeline.
10 points - Highlighted complexity: Built and optimized the transformer from scratch.
10 points - Exceptional visualization/diagrams/repo: Created project architecture diagram and a clean README which includes steps to run and explanation of folder structure.

10 points - Discussion of lessons learned and potential improvements: This project taught me a lot about transformers. One of the key lessons I learned was the importance of tailoring the model architecture to the problem. An architecture that works for translation doesn't automatically work for summarization. I also realized how critical preprocessing is; cleaning and normalizing the input text had a direct impact on how well the model could learn and generate coherent summaries.

If I were to improve the project, I'd experiment with pretrained models like BART or T5 to take advantage of their language understanding, especially given the limited dataset. I'd also add better evaluation metrics like ROUGE to track performance more objectively. Overall, this was a valuable deep dive into both the potential and the challenges of applying transformers to real-world NLP tasks.

Micah: Also worked on the transformer model and modified it to produce review summaries. Designed and built the user-facing application to improve accessibility and presentation of results. Also worked on the demo.
80 points - Significant exploration beyond baseline: Adjusted initial transformer to work with reviews for summarization, experimented with transformer architecture to optimize output, learned streamlit for GUI for the program.
30 points - Innovation or Creativity: Trained a transformer to output summaries of Amazon product reviews. Created a working Streamlit-based graphical user interface (GUI) despite having no prior experience with the framework. Designed the GUI to clearly present summaries, pros/cons, and gender distribution visualizations to users.
10 points - Highlighted complexity: Built and optimized the transformer from scratch.
10 points - Discussion of lessons learned and potential improvements: One key takeaway was understanding the importance of preprocessing in real-world NLP tasks. Review text often contained inconsistencies, slang, and grammatical errors, all of which negatively affected the model's coherence. By improving the preprocessing steps, the model was able to produce clearer, more reliable summaries. For future improvements, training with more diverse and cleaner review datasets would enhance model performance.
10 points - Exceptional visualization/diagrams/repo: Designed a streamlit app with interactive components to showcase the models rather than running a Python file.

Nathaniel: Sourced dataset for the project. Worked on the pros/cons list generator and the gender distribution analysis.
80 points - Significant exploration beyond baseline: Experimented with different NLP techniques for sentiment analysis and gender prediction, and compared their outputs to devise the best solution
30 points - Innovation or Creativity: In response to difficulties with sentiment analysis, I combined POS-tagging with rating data to refine the pros/cons generation and output more descriptive information to users. This was a unique solution to gather product insights by using the data in a creative way that avoided making the solution unnecessarily complex.

10 points - Highlighted complexity - Sourced the dataset used for the project, researched various methods for generating pros and cons, tested the gender prediction model on various username examples to understand how different types of names get classified.
10 points - Discussion of lessons learned and potential improvements: I learned that trying to force a complicated technique, in this case sentiment analysis, is not always the best solution. I would've saved myself a lot of time if I used the resources I had available to me from the beginning, such as the "Rating" column. It already had data that closely corresponded to sentiment, so using the ratings instead of analyzing the sentiment and using that to determine whether something was a pro and con resulted in more accurate results.

Further improvements could be made by expanding the types of demographics that were predicted. Features such as age, race, or geography would be useful information for users and vendors to have, and could likely be similarly extracted from reviews or profile names. Further research into aspect-based sentiment analysis could be used to further refine the pros/cons list, especially in cases of mixed sentiment. Finding or creating a bigger dataset with more reviews would also make the cons list more accurate, as there were few negative reviews for each given product.
10 points - Exceptional visualization/diagrams/repo: Gender distribution is visualized as a pi chart, and formatted the pros/cons into dataframes for better visualization.
10 points - Discussion of testing outside of the team, on 5 people: After developing the core components of our NLP project, we invited various potential users and friends to test the system in order to identify any usability issues and gather feedback. After initial testing, we identified several usability issues with the system. Testers struggled with navigating the results due to the lack of a clear, user-friendly interface within the Python file. To address this, we developed an application with an approachable interface that made the data more accessible, and displayed clear summaries and visualizations.

Another major feedback point was that some pros and cons didn't make sense. This was when we were still tagging nouns as potential pros and cons, and would generate pros such as "the cookies," which were the products themselves. This caused us to experiment more with this method and ultimately switched to only extracting adjectives from the reviews, which later testers found more descriptive.

Citations

[1] Vaswani, Ashish, et al. "Attention Is All You Need." *arXiv.Org*, 2 Aug. 2023, arxiv.org/abs/1706.03762.

[2] Name-Gender-Predictor by Imshibl.